



# John Vlissides' biases

---

- Incremental progress over breakthroughs
  - lots of little synergies
- Can't divorce
  - in-large & in-small programming
  - specification, design, & implementation
- Outside examples
  - "co-design," "concurrent engineering," "platform teams," ...



# Language advancements

---

- Expressiveness
  - mechanisms, idioms, sep of concerns
- Transparency
  - location/platform-indep, semiauto customization
- Adaptability
  - reflection, BCA
- Robustness
  - adv exceptions, asserts/constraints, built-in testability



# Architecture & design

---

- Becomes a discipline
  - literature, standards, accountability
- Patterns as literature
  - managing their explosion
  - integration into process



# Development tools

---

- Advanced environments
  - slicing, concurrency/distrib savvy, code customization, refactoring, collaboration
- Modeling & simulation
  - behavior, trade time/accuracy, estimation
- Run-time analysis
  - visualization, trace-driven optimization, code partitioning



# Summary

---

- *Incremental* advances in prog langs
  - new mechs + institutionalization of idiomatic code
  - most new capabilities are in libraries
- *Fundamental* advances in dev tools
  - collaborative support, run-time analysis
  - design-driven tools
- *Greatest* advances in software arch & design
  - emerging discipline
  - greatest potential leverage